

Question 1

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Ingatkah Anda dengan Praktikum Alstrukdat? dengan Queue? Selamat di datang di praktikum OOP minggu ini. Kali ini silahkan implementasikan Priority Queue dengan memanfaatkan Generic Class dan STL berupa Vector!

Lengkapi [PriorityQueue.hpp](#) berikut dan kumpulkan kembali PriorityQueue.hpp yang telah berisi implementasi dari fungsi-fungsinya. Priority queue merupakan queue yang elemennya terurut dari elemen terbesar ke terkecil sehingga proses enqueue akan menempatkan elemen baru ke posisi yang sesuai sehingga keterurutan dari priority queue tetap terjaga.

Penjelasan STL Vector

Vector adalah kontainer data berupa array dinamis yang dapat mengubah ukurannya ketika dilakukan penambahan elemen baru ataupun penghapusan elemen. Pada dasarnya, Vector sama seperti array.

Vector mempunyai beberapa method dasar seperti

1. **bool empty()** – Mengembalikan apakah vector sedang kosong.
2. **size_type size()** – Mengembalikan ukuran vector.
3. **void pop_back()** – Menghapus elemen paling kanan pada vector.
4. **void push_back(value_type val)** – Menambahkan elemen 'val' pada vector.

Berikut adalah contoh penggunaan STL vector

```
1  #include <vector>
2  #include <iostream>
3
4  using namespace std;
5
6  int main(){
7      vector<int> vector;
8
9      vector.push_back(1);
10     vector.push_back(2);
11     vector.push_back(3);
12     vector.push_back(4);
13     vector.push_back(5);
14
15     vector.pop_back();
16
17     for(int i = 0; i < vector.size(); i++){
18         cout << vector[i] << " ";
19     }
20 }
```

C++14

 [PriorityQueue.hpp](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 2.95 MB
2	10	Accepted	0.00 sec, 2.95 MB
3	10	Accepted	0.00 sec, 2.99 MB
4	10	Accepted	0.00 sec, 2.99 MB
5	10	Accepted	0.00 sec, 3.00 MB
6	10	Accepted	0.00 sec, 2.90 MB
7	10	Accepted	0.00 sec, 3.00 MB
8	10	Accepted	0.00 sec, 2.90 MB
9	10	Accepted	0.00 sec, 3.07 MB
10	10	Accepted	0.00 sec, 2.92 MB

Question **2**

Correct

Mark 100.00 out of 100.00

Time limit	1 s
Memory limit	64 MB

Tuan Haka adalah seorang walikota di Bondoland. Dia adalah seorang walikota yang teliti dalam pekerjaannya. Dia selalu mampu memeriksa proyek bangunan yang akan berjalan tanpa kesalahan. Namun, dia bertambah tua dan sering merasa kelelahan karena pekerjaannya yang sangat banyak. Ia meminta bantuan developer handal untuk membantunya membuat program yang dapat membangun bangunan sesuai dengan spesifikasi. Program harus bisa memberikan pesan kesalahan ketika ada syarat yang tidak terpenuhi.

Berikut adalah file header yang diberikan:

- [Exception.hpp](#)
- [Bangunan.hpp](#)
- [Walikota.hpp](#)

Anda perlu membuat **Bangunan.cpp** dan **Walikota.cpp** sesuai spesifikasi yang diberikan.

Catatan:

Semua output diakhiri dengan newline

C++14

 [Walikota.zip](#)

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 2.99 MB
2	10	Accepted	0.01 sec, 2.95 MB
3	10	Accepted	0.00 sec, 3.04 MB
4	10	Accepted	0.01 sec, 2.89 MB
5	10	Accepted	0.00 sec, 2.92 MB
6	10	Accepted	0.00 sec, 2.89 MB
7	10	Accepted	0.00 sec, 2.94 MB
8	10	Accepted	0.00 sec, 2.88 MB
9	10	Accepted	0.00 sec, 2.94 MB
10	10	Accepted	0.00 sec, 2.98 MB

Time limit	1 s
Memory limit	64 MB



Tuan Haka merasa pekerjaannya sebagai walikota di Bondoland kurang menantang. Untuk itu, ia memutuskan untuk membuka restoran di Bondoland. Restoran ini menjual berbagai menu makanan dan dalam 1 kali order, customer dapat memesan lebih dari 1 menu makanan. Agar pekerjaannya dapat berjalan lancar, ia butuh bantuan Anda untuk membuat sebuah program yang membantu penerimaan order makanan pada restoran ini. Order makanan memanfaatkan data struktur **stack**.

Berikut adalah file header yang diberikan:

1. [Restaurant.hpp](#)

Anda perlu membuat **Restaurant.cpp** sesuai spesifikasi yang diberikan. Jangan lupa newline pada setiap output.

Penjelasan STL Stack

Stack adalah kontainer data dengan mekanisme LIFO (Last In First Out). Elemen paling puncak pada stack disebut sebagai TOP. Penyisipan pada stack selalu dilakukan “di atas” TOP dan penghapusan selalu dilakukan pada TOP. TOP adalah satu-satunya alamat tempat terjadi operasi.

Stack mempunyai beberapa method dasar seperti

1. **bool empty()** – Mengembalikan apakah stack sedang kosong.
2. **size_type size()** – Mengembalikan ukuran stack.
3. **reference top()** – Mengembalikan reference kepada element TOP pada stack.
4. **void push(value_type g)** – Menambahkan elemen 'g' diatas TOP pada stack.
5. **void pop()** – Menghapus element TOP pada stack.

Berikut adalah contoh penggunaan STL stack

```
1 #include <iostream>
2
3 #include <stack>
4
5 using namespace std;
6
7 int main() {
8
9     stack<int> stack;
10
11     stack.push(21);
12
13     stack.push(22);
14
15     stack.push(24);
16
17     stack.push(25);
18
19     int num=0;
20
21     stack.push(num);
22
23     stack.pop();
24
25     stack.pop();
26
27     stack.pop();
28
29
30
31     while (!stack.empty()) {
32
33         cout << stack.top() << " ";
34
35         stack.pop();
36
37     }
38
39 }
```

Score: 100

Blackbox

Score: 100

Verdict: Accepted

Evaluator: Exact

No	Score	Verdict	Description
1	10	Accepted	0.00 sec, 2.89 MB
2	10	Accepted	0.00 sec, 3.00 MB
3	10	Accepted	0.00 sec, 3.01 MB
4	10	Accepted	0.05 sec, 2.89 MB
5	10	Accepted	0.00 sec, 2.93 MB
6	10	Accepted	0.00 sec, 2.89 MB
7	10	Accepted	0.00 sec, 2.91 MB
8	10	Accepted	0.00 sec, 2.98 MB
9	10	Accepted	0.00 sec, 2.96 MB
10	10	Accepted	0.00 sec, 2.92 MB

[← Praktikum 3](#)

Jump to...

[UTS Praktikum ►](#)